

# Lista de Exercícios - Métodos de busca e comparação de algoritmos

BCC702 - Programação de Computadores II

2025.1

## INSTRUÇÕES

- A atividade é individual.
  - Implemente **TODOS** os exercícios.
  - Desenvolver o código e realizar os testes em C++.
  - Incluir comentários no código explicando as etapas. Comente trechos do código, não linha a linha!
  - Envie suas soluções no Moodle. Para enviar, crie um arquivo compactado com todas as suas soluções e envie o arquivo compactado.
    - Para questões que criem algum arquivo, não é necessário enviar o arquivo gerado.
  - Exercícios enviados fora do prazo serão penalizados com 10% da nota por dia de atraso.
- 

## Exercício 1

Considere a implementação do algoritmo Busca Sequencial apresentada na aula. Faça uma versão deste algoritmo, trocando a estrutura de repetição **for** por **while**.

Para os testes leve em consideração o vetor: [1, 5, 9, 7, 12, 3, 17]

### Exemplo 1

Digite qual valor deseja procurar: 7

Valor encontrado!

### Exemplo 2

Digite qual valor deseja procurar: 4

Valor não encontrado!

## Exercício 2

Faça um programa que use o método de busca binária para encontrar e imprimir a posição de um determinado elemento no vetor. Se o elemento não existir, seu programa deve imprimir o valor -1.

O tamanho e os dados do vetor serão informados pelo usuário, então lembre-se de alocar dinamicamente (e liberar) o vetor.

**Não se esqueçam que o vetor deve estar ordenado.** Não é necessário ordenar o vetor, apenas tome cuidado ao inserir os dados.

### Exemplo 1

Digite o tamanho do vetor: 5

Digite os elementos do vetor: 1 2 3 4 5

Digite o elemento procurado: 3

Posição do elemento procurado: 2

### Exemplo 2

Digite o tamanho do vetor: 7

Digite os elementos do vetor: 4 8 7 11 17 28 31

Digite o elemento procurado: 3

Posição do elemento procurado: -1

## Exercício 3

Altere o método de busca binária para que, a cada divisão do vetor, ele imprima quantos elementos foram descartados na busca.

Você pode utilizar o código da questão anterior como base.

### Exemplo 1

Digite a quantidade de elementos do vetor: 5

Digite os elementos do vetor: 1 2 3 4 5

Digite o elemento procurado: 1

1ª iteração: 3

2ª iteração: 1

## Exercício 4

A função a seguir implementa a versão recursiva da busca binária. Complete os valores das lacunas.

```
bool buscaR(int v[], int n, int procurado, int esq, int dir){
    int meio = (esq+dir)/2;
    if (esq <= dir){
        if(v[meio] == procurado)
            return true;
        else if(v[meio] > procurado)
            return buscaR(v, n, procurado, -----, -----);
        else if(v[meio] < procurado)
            return buscaR(v, n, procurado, -----, -----);
    }
    return false;
}
```

Agora responda:

1. Qual é o melhor caso dessa função? Em um vetor de tamanho  $n$ , quantas comparações ocorreriam?
2. Qual é o pior caso dessa função? Em um vetor de tamanho  $n$ , quantas comparações ocorreriam?
3. Em qual classe de algoritmos (reveja a tabela apresentada na aula) essa função se encaixa? Justifique.

## Exercício 5

Análise a função a seguir:

```

int descubra(int vet[], int n, int k){
    int x = 0, y = 0;
    for(int i = 0; i < n; i++){
        if(vet[i] < k) { // comparacao 1
            x += vet[i];
        }
        if(vet[i] > k) { // comparacao 2
            y += vet[i];
        }
    }
    return y-x;
}

```

1. O que essa função retorna?
2. Quantas vezes a comparação 1 é realizada?
3. Quantas vezes a comparação 2 é realizada?
4. Qual é o total de comparações?